

CASE NO.: HSJ920030256US1  
Serial No.: 10/756,123  
March 15, 2007  
Page 2

PATENT  
Filed: January 12, 2004

"practical application." But the rejection is merely conclusory. It utterly fails to explain why a GUI, much less one that is explicitly disclosed to be displayed on a computer monitor for configuring a useful, tangible, and concrete data pipeline, lacks a practical application when it so obviously is highly practical. Applicant's prior comments in this regard are incorporated herein to preserve them for appeal.

#### Obviousness Rejections - All Claims

Claims 1-9 have been rejected under 35 U.S.C. §103 as being unpatentable over ref (a), alleging that paragraphs 66 line 12 through paragraph 67 line 11 teach a pipe input set window configured to permit a user to define a type of pipe input set data and that paragraph 78 teaches producing a GUI by translating the type using a configuration file to a class, in view of ref (b), alleging that col. 5, lines 56-60 teach Java reflection.

Turning first to ref (a), the relied-upon portions state nothing at all about defining an input set data type as claimed. Instead, in pertinent part the relied-upon portion of paragraph 66 teaches simply that a node in the graph represents a specific predefined data transformation functionality that is offered by uninstantiated component objects 370 residing in a component library 316 - not that a *type* of a data set, as opposed to a transformation, is defined, and less still that the GUI page is generated by translating the type using a configuration file to a class as claimed.

The relied-upon portion of paragraph 67 is of no further avail, because it too fails to state anything about defining a type for an input data set. Instead, it merely teaches that after graph data is input via the GUI, a translator translates the graph into an DFE plan. The translator also works in conjunction with an optimizer subsystem to optimize the graph into a maximally efficient execution structure by eliminating

CASE NO.: HSJ920030256US1  
Serial No.: 10/756,123  
March 15, 2007  
Page 3

PATENT  
Filed: January 12, 2004

redundancies, simplifying and enhancing the DFE plan and possibly performing other optimizations - but not by defining a type of an input data set. For this reason, the rejections are overcome.

In addition, Applicant would like to point out that paragraph 78 of ref (a) does not, contrary to the allegation in the rejection, produce a GUI by translating the type using a configuration file to a class. Nothing in paragraph 78 implicates a type of an input data set, much less to use the type to produce a GUI, much less still in the particular way claimed. Instead, paragraph 78 further discusses the importance of the translator/optimizer portions discussed above (which are parts of a "scheduler"), and that the scheduler further controls actual execution of data flows. For this further reason, the rejections are overcome.

Still further, the suggestion to modify ref (a) with the alleged "Java reflection" of ref (b) to arrive at Claim 1 is misplaced. Specifically, the relied-upon portion of ref (b) simply states that each Java bean component has its own features, including its properties, methods, and events, and that the features can be examined by other beans by a process known as introspection, which is supported in two ways. First, ref (b) teaches that specific rules, known as design patterns, are followed when naming bean features. Ref (b) explains that a particular class examines beans for these design patterns to discover bean features, and that this class relies on the "core reflection API." But this API is never further mentioned, either in terms of what it is or what it does.

Accordingly, to the extent that the "core reflection API" is the same thing as "Java reflection", which Applicant does not concede, it is used at most to examine the features of Java beans. In contrast, Claim 1 requires using Java reflection for something entirely different and, hence, unsuggested by ref (b), namely, generating an instance of a class of a type of pipe input set data. Thus, ref (b) does not suggest anything remotely close to a modification of ref (a) that would teach Claim 1, further rendering the rejection overcome.

1189-25.AM2

CASE NO.: HSJ920030256US1  
Serial No.: 10/756,123  
March 15, 2007  
Page 4

PATENT  
Filed: January 12, 2004

#### Obviousness Rejections - Additional Comments Regarding Dependent Claims

With respect to Claim 2, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraph 82 teaches a pipe input set window and GUI page that require no programming apart from an initial core code. All this part of ref (a) teaches is the details of how the translator/optimizer translate the graph into a DFE.

With respect to Claim 3, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraph 82 teaches that the GUI is an incremental GUI wherein GUI pages for new pipe components can be added incrementally without changing existing code. All this part of ref (a) teaches is the details of how the translator/optimizer translate the graph into a DFE.

With respect to Claim 4, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraph 9 teaches that a new pipe module is based on a pre-existing module type. All this portion of ref (a) teaches is that GUI graph can be used to develop a DFE, with each node in the graph defining a transformation function.

With respect to Claim 6, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraphs 69-73 teach that the GUI defines a set of interfaces, with each interface including plural functions. The relied-upon paragraphs do not describe interfaces much less ones that includes plural functions, but rather simply set out the five operations used to control the DFE of ref (a).

With further respect to Claim 6, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraphs 53 and 81 teach that the GUI includes a GUI representation part and a storage part, with the GUI representation part defining how something is displayed and the storage part defining how GUI parameters are stored in an external storage. Instead, paragraph 53 discusses exposing a user interface

CASE NO.: HSJ920030256US1  
Serial No.: 10/756,123  
March 15, 2007  
Page 5

PATENT  
Filed: January 12, 2004

to an SQL server, not that a GUI includes two parts, much less the ones claimed, much less still what the parts define. Paragraph 81 likewise adds nothing of import, describing only how the GUI is used to define how to extract data from a database.

With respect to Claim 7, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraph 80 teaches a tab for defining a set representative of a type of output data from the pipeline. Paragraph 80 merely states that data remains in a buffer during DTP operations. No tab is mentioned, much less in the context of a GUI, much less still for the purpose recited in Claim 7.

With respect to Claim 8, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraph 78 teaches a tab for defining an arbitrary number of elements contained in a component of the pipeline, with individual input and output sets being definable for each element in the component. Instead, paragraph 78 further discusses the importance of the translator/optimizer portions discussed above (which are parts of a "scheduler"), and that the scheduler further controls actual execution of data flows. There is no teaching in paragraph 78 of a tab, arbitrary number of elements in a component, or individual input and output sets that are defined for each element in the component.

With respect to Claim 9, Applicant would like to point out that the allegation appears to be incorrect that ref (a), paragraph 7 teaches a tab for defining an arbitrary number of modules of the pipeline, with a type being selected for each module using the tab and with the type defining part of the GUI. Instead, paragraph 7 contains no teachings of tabs, much less tabs that are part of a GUI, or of selecting modules, much less doing so using GUI tabs. Paragraph 7 simply discusses scheduling data flow execution using a graph.

1119-25.AMC

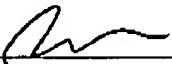
FROM ROGITZ 619 338 8078

(THU)MAR 15 2007 8:13/ST. 8:12/No. 6833031659 P 6

CASE NO.: HSJ920030256US1  
Serial No.: 10/756,123  
March 15, 2007  
Page 6

PATENT  
Filed: January 12, 2004

Respectfully submitted,

  
John L. Rogitz  
Registration No. 33,549  
Attorney of Record  
750 B Street, Suite 3120  
San Diego, CA 92101  
Telephone: (619) 338-8075

JLR:jg

1189-25.AMZ